https://doi.org/10.31436/iiumej.v23i1.1783

IN-IDRIS: MODIFIKASI STEMMING IDRIS ALGORITMA TEKS BAHASA INDONESIA

FEBIARTIWULANSUCI, NKAMUHAYATIN*DANYUDAMUNARKO

Jurusan Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang, Malang, Indonesia

**Penulis koresponden: noorhayatin@umm.ac.id*

(Diterima: 11thJanuari 2021; Diterima: 18thMaret 2021; Diterbitkan secara online: 4thJanuari 2022)

ABSTRAK: Stemming mempunyai peranan penting dalam pengolahan teks. Stemming setiap bahasa berbeda-beda dan sangat dipengaruhi oleh jenis teks bahasanya. Selain itu setiap bahasa mempunyai aturan yang berbeda-beda dalam penggunaan kata dengan an*afiks*. Sebagian besar kata yang digunakan dalam bahasa Indonesia dibentuk dengan menggabungkan akar kata dengan*imbuhan* dan bentuk gabungan lainnya. Salah satu permasalahan dalam stemming di Indonesia adalah adanya perbedaan jenis*imbuhan*, dan juga memiliki beberapa*awalan*yang berubah sesuai dengan huruf pertama dari kata dasar. Penerapan stemmer Idris untuk teks bahasa Indonesia menarik karena Indonesia dan Malaysia mempunyai akar bahasa yang sama. Namun hasilnya tidak selalu menghasilkan kata yang sebenarnya, karena algoritma Idris terlebih dahulu menghilangkan awalan sesuai Rule 2. Penghapusan ini secara langsung mempengaruhi hasil stemmer Idris ketika diimplementasikan ke teks bahasa Indonesia. Dalam penelitian ini, kami fokus pada modifikasi stemmer Idris (dari bahasa Melayu) menjadi IN-Indris dengan konteks Indonesia. Untuk menguji usulan modifikasi algoritma asli, kutipan novel online Indonesia digunakan untuk mengukur kinerja IN-Idris. Pengujian dilakukan untuk membandingkan algoritma yang diusulkan dengan algoritma lainnya*stemmer*. Dari hasil percobaan, IN-Idris mempunyai akurasi kurang lebih 82,81%. Terjadi peningkatan akurasi hingga 5,25% jika dibandingkan dengan akurasi Idris. Apalagi stemmer yang diusulkan juga berjalan lebih cepat dari Idris dengan selisih kecepatan sekitar 0,25 detik.

ABSTRAK: Stemming mempunyai peranan penting dalam pemprosesan teks. Batang setiap bahasa berbeza dan sangat dipengaruhi oleh jenis teks bahasa. Selain itu, setiap bahasa mempunyai peraturan yang berbeza dalam penggunaan kata dengan awalan. Sebilangan besar kata-kata yang digunakan dalam bahasa Indonesia dibentuk dengan menggabungkan kata akar dengan afiks dan bentuk gabungan lainnya. Salah satu masalah dalam bahasa Indonesia adalah mempunyai pelbagai jenis awalan, dan juga mempunyai beberapa awalan yang berubah sesuai dengan huruf pertama kata dasar. Penerapan stemder Idris untuk teks Indonesia adalah minat kerana Indonesia dan Malaysia mempunyai akar bahasa yang sama. Namun, hasilnya tidak selalu menghasilkan kata yang sebenarnya, karena algoritma Idris pertama kali Menghapus awalan menurut Peraturan 2. Penghapusan ini secara langsung mempengaruhi hasil batang Idris ketika diterapkan ke teks Indonesia. Dalam kajian ini, kami fokus pada stemmer Idris yang diubahsuai (dari bahasa Melayu) menjadi IN-Indris dengan konteks Indonesia. Untuk menguji cadangan pengubahsuaian pada algoritma asli, petikan novel dalam talian Indonesia digunakan untuk mengukur prestasi IN-Idris. Pengujian dilakukan untuk memetakan algoritma yang dicadangkan dengan stemmer lain. Dari hasil eksperimen, IN-Idris mempunyai presisi presisi sekitar 82,81%, terdapat akurasi presisi hingga 5,25% dibandingkan dengan presisi presisi Idris. Selain itu, stemmer yang dicadangkan juga berjalan lebih cepat daripada Idris dengan jurang kelajuan sekitar 0.25 saat.

KATA KUNCI:*Idris bertangkai; DI-Idris; NLP; pra-pemrosesan teks*

1. PERKENALAN

Menemukan akar kata dari kata turunan masih menjadi tantangan, terutama untuk pengolahan teks. Stemming adalah proses mengubah kata turunan menjadi kata dasar. Metode stemming banyak diterapkan dalam Natural Language Processing (NLP) dan juga text mining terutama ketika terlibat dalam penggalian informasi yang digunakan untuk mencari informasi baru dari berbagai sumber tertulis [1], seperti analisis sentimen [2]. Stemmer, salah satu algoritma stemming, mempunyai peranan penting dalam pengolahan teks karena hasil stemming digunakan untuk mengekstrak fitur-fitur yang ada pada teks. Untuk area text mining, stemmer penting pada tahap awal, yang memiliki fungsi untuk mengubah teks tidak terstruktur menjadi format representatif terstruktur yang kemudian dapat diproses oleh mesin [3]. Penerapan stemmer untuk pemrosesan teks terbukti meningkatkan kinerja pra-pemrosesan teks [4]. Hal ini juga berdampak pada hasil klasifikasi contoh menggunakan terjemahan Al-Quran bahasa Indonesia [5], dan mereduksi perbedaan bentuk kata menjadi satu akar kata [6].

Stemming setiap bahasa berbeda-beda, dan hal ini sangat dipengaruhi oleh jenis teks bahasanya. Apalagi setiap bahasa mempunyai aturan berbeda dalam penggunaan kata dengan*imbuhan* [7]. Namun, untuk beberapa bahasa, stemming dapat dicapai dengan menerapkan aturan morfologi yang sesuai [8]. Bahasa Indonesia menggunakan Bahasa Indonesia sebagai bahasa formal. Dalam teks bahasa Indonesia, kata terbentuk dari aturan morfologi [9]. Sebagian besar kata yang digunakan dalam bahasa Indonesia dibentuk dengan menggabungkan akar kata dengan*imbuhan* dan bentuk penggabungan lainnya [10]. Bahasa Indonesia mempunyai permasalahan yang bersifat khusus pada bahasa tersebut. Salah satu masalahnya adalah memiliki jenis yang berbeda*afiks*, yang lain sedang makan*awalan*yang berubah sesuai dengan huruf pertama dari kata dasar. Misalnya, *awalan*"*Saya*-" menjadi "*mem*-" bila dilampirkan pada akar kata yang dimulai dengan huruf "*B*-" seperti dalam "*mem-buat*" (untuk membuat, dalam bahasa Inggris), tetapi menjadi "*tidak*-" bila dilampirkan pada akar kata yang dimulai dengan huruf "*S*-" seperti dalam "*tidak*-[S] *impan*" (untuk menyimpan, dalam bahasa Inggris) [11].

Dalam pengolahan teks bahasa Indonesia, beberapa peneliti menggunakan algoritma stemming seperti Nazief dan Adriani (N&A) [12], Confix Stripping Algorithm (CS) [13], dan Enhanced Confix Stripping (ECS) [14], yang dikembangkan untuk teks bahasa Indonesia . Namun beberapa peneliti juga menerapkannya*stemmer*dari bahasa lain seperti Idris stemmer (untuk bahasa Melayu) [15], dan Potter stemmer (untuk bahasa Inggris) [16]. Dalam penelitian ini, kami fokus pada Idris stemmer. Hal ini menarik karena bahasa Indonesia dan Melayu mempunyai akar bahasa yang sama. Bahasa Melayu mempunyai kemiripan dengan bahasa Indonesia, sehingga algoritma Idris juga dapat digunakan untuk mengolah teks bahasa Indonesia [17].

Penelitian sebelumnya membandingkan waktu pemrosesan dan keakuratan stemmer Nazief dan Adriani (N&A) dengan stemmer Idris [8]. Penelitian ini diterapkan pada teks bahasa Indonesia untuk mengetahui algoritma mana yang lebih baik. Untuk evaluasi, peneliti menggunakan lima teks cerita untuk membandingkan hasil stemming. Hasilnya menunjukkan bahwa stemmer N&A memperoleh akurasi sebesar 97% dengan waktu pemrosesan sekitar 0,03 detik per kata, sedangkan stemmer Idris mencapai akurasi 91% namun memerlukan waktu pemrosesan sekitar 0,02 detik per kata. Penelitian lain berfokus pada analisis kekuatan stemmer pada dokumen teks bahasa Indonesia berdasarkan parameternya*ic*Dan*toilet*nilai, dan juga menganalisis tingkat akurasi dan kecepatan berdasarkan hasil kata. Hasilnya, Idris stemmer sudah akurat dan berhasil menghasilkan akar kata dari teks bahasa Indonesia, namun masih mempunyai kekurangan yaitu, memperbesar kemungkinan*berlebihan*. Hasil stemming kurang sesuai karena algoritma Idris terlebih dahulu menghilangkannya*awalan*menurut Aturan 2. Eliminasi langsung mengacu pada efek Aturan 2 pada hasil stemming, dimana

https://doi.org/10.31436/iiumej.v23i1.1783

algoritma tidak selalu menghasilkan kata yang sebenarnya. Selain itu, percobaan lain [8] telah membuktikan bahwa algoritma Idris mempunyai keunggulan pada kecepatan proses stemming namun tingkat akurasinya masih rendah.

Untuk memperoleh tingkat akurasi yang lebih baik dan proses stemming yang lebih cepat, penelitian ini mengusulkan untuk memodifikasi algoritma Idris dengan teks bahasa Indonesia. Kami mengukur keakuratan dan kecepatan proses stemming sebagai hasil modifikasi algoritma Idris. Algoritma yang dimodifikasi disebut IN-IDRIS.

2. PEKERJAAN TERKAIT

2.1 Algoritma Stemmers Indonesia

Stemmer Nazief dan Adriani (N&A) [12] didasarkan pada aturan morfologi yang saling terkait dan dikelompokkan, kemudian merangkum bagian kata yang diperbolehkan dan tidak menyertakan *imbuhan*seperti*awalan*,*akhiran*, Dan*konfigurasi*untuk mendapatkan akar kata. Kinerja algoritma ini didasarkan pada tiga bagian yang dikelompokkan*imbuhan*, aturan penggunaan dan penetapan batasan, serta penggunaan kamus. Kamus menjadi bagian yang penting karena digunakan untuk memeriksa apakah suatu kata sudah memenuhi kata dasar atau belum. Sebelum*afiks* Dalam proses penghapusan, beberapa hal yang harus diperhatikan dalam penggunaan algoritma ini, seperti *akhiran infleksi*, derivasi*akhiran*, derivasi*awalan*, Dan*awalan*tidak diizinkan*akhiran* [18]. Pendekatan kinerja N&A adalah pendekatan yang paling kompleks.

Algoritma Confix Stripping (CS) merupakan stemmer bahasa Indonesia [6] yang dikembangkan berdasarkan penyempurnaan dari algoritma Nazief & Adriani. CS stemmer menambahkan langkah stemming tambahan yang disebut*LoopPengembalianAkhiran*(putaran pengembalian terakhir). Algoritma ini dilakukan dengan menambahkan beberapa*awalan*aturan dan modifikasi*awalan*aturan, khususnya menambahkan prioritas aturan. Dari perbandingan hasil kinerja menunjukkan bahwa algoritma CS mempunyai kinerja yang lebih baik dibandingkan dengan N&A stemmer [13]. Perkembangan dari CS stemmer adalah Enhanced Confix stripping (ECS) [14]. Dari percobaan yang dilakukan diperoleh hasil bahwa ECS stemmer berhasil meningkatkan akurasi dari modifikasi ECS stemmer menggunakan metode non deterministik yang mampu mengidentifikasi kemungkinan-kemungkinan akar kata yang dapat dibentuk dalam satu kata melalui kandidat. daftar [19].

Stemmer bahasa Indonesia berdasarkan kamus adalah stemmer VEGA. Algoritma ini menggunakan kumpulan aturan untuk menentukan apakah suatu*afiks*dapat dihilangkan dari sebuah kata. Aturan diakses sesuai urutan penyajiannya dalam kode. Ketika satu aturan gagal, algoritma melanjutkan ke aturan berikutnya. Kelemahan utama pendekatan VEGA adalah tidak adanya tahap pencarian di mana kata-kata hanya dibandingkan dengan kamus akar kata yang diketahui. Stemming berlanjut selama kata tersebut mengandung*afiks*huruf, sering kali mengarah ke *berlebihan*. Selain itu, algoritma ini tidak melayani kasus-kasus di mana pengodean ulang diperlukan. Yang terakhir, ketergantungan pada peraturan yang ketat mengharuskan peraturan tersebut benar dan lengkap, dan mencegah pemulihan yang bersifat ad hoc*afiks*kombinasi [20].

2.2 Algoritma Idris Stemmer

Stemmer Idris dikembangkan oleh Idris dkk. yang dirancang untuk bahasa Melayu. Bahasa Melayu mempunyai kemiripan dengan bahasa Indonesia [16] sehingga algoritma ini juga berlaku untuk teks bahasa Indonesia. Algoritma ini memiliki dua kamus (kamus umum dan kamus lokal) dalam menentukan hasil stemming [17]; dimana kamus lokal memuat daftar akar kata dalam kosakata sejarah Melayu. Algoritma ini hanya mengimplementasikan dua pola aturan yaitu*awalan*Dan*akhiran*aturan. Dengan hanya menggunakan dua pola *imbuhan*itu adalah*awalan*Dan*akhiran*, hal ini dapat mengurangi jumlah kumpulan aturan.

https://doi.org/10.31436/iiumej.v23i1.1783

Algoritma Idris pertama-tama memeriksa kata-kata tersebut*awalan*aturan kemudian memeriksa kata-kata terhadap*akhiran*aturan. Jika tidak, banyak kesalahan seperti*berlebihan*dapat terjadi [15]. Algoritma Idris mengadopsi asumsi Arifin dan Setiono bahwa setiap kata dalam bahasa Indonesia mempunyai dua*awalan* dan tiga*akhiran*[21].

Algoritma Idris menerapkan skema pengodean ulang dan stemming progresif yang berbeda. Algoritme memeriksa kamus setelah setiap langkah, menghentikan dan mengembalikan kata dasar jika ditemukan. Skema ini bekerja sebagai berikut: pertama, setelah memeriksa kata dalam kamus, algoritma menguji apakah kata tersebut*awalan*kata yang cocok dengan a*awalan*yang mungkin memerlukan pengodean ulang; kedua, bila diperlukan pengodean ulang, dilakukan dan dicari kata yang dihasilkan di kamus, sedangkan bila tidak diperlukan pengodean ulang, dilakukan pencarian kata yang dihasilkan.*awalan*dihapus seperti biasa dan kata tersebut diperiksa di kamus; ketiga, setelah menghapus a*awalan*,*akhiran*penghapusan dicoba dan, jika berhasil, kata tersebut diperiksa di kamus; dan terakhir, algoritma kembali ke langkah kedua dengan kata yang berasal sebagian. Ada dua varian dari algoritma ini: perubahan pertama*awalan*dan kemudian melakukan pengodean ulang, sedangkan yang kedua melakukan sebaliknya [22]. Deskripsi algoritma Idris sebagaimana dimaksud pada Tabel 1.

Tabel 1. Deskripsi algoritma Idris

ALGORITMA 1 : Algoritma Idris
1. Periksa kata tersebut di kamus. Apabila kata tersebut ditemukan, maka kata tersebut dianggap sebagai kata dasar dan kata keluar. Jika tidak, lanjutkan ke langkah berikutnya.
2. Periksa kata di*awalan*aturan. Jika kata tersebut cocok dengan*awalan*aturan, periksa *awalan*pola dan huruf pertama kata dasar. Jika tidak, lanjutkan ke langkah 7.
3. Jika*awalan*polanya cocok dengan pola pada Aturan 2, lalu terapkan Aturan 2 pada kata tersebut. Jika tidak, hapus*awalan*dan lanjutkan ke langkah 6.
4. Periksa*awalan*dari kata tersebut, sesuaikan polanya pada Aturan 2. Jika cocok dengan aturan keempat, maka periksa kata dasar di kamus dan lanjutkan ke langkah berikutnya. Jika tidak, hapus*awalan*dan lanjutkan ke langkah 6.
5. Jika kata tersebut tidak ada dalam kamus, kembali ke langkah 4, jika tidak, hapus*awalan* dan lanjutkan ke langkah berikutnya.
6. Periksa kata tersebut di kamus. Apabila kata tersebut ditemukan, maka kata tersebut dianggap sebagai kata dasar dan kata keluar. Jika tidak, lanjutkan ke langkah berikutnya.
7. Periksa kata-kata di*akhiran*aturan. Jika cocok dengan*akhiran*aturan, lalu hapus

*Awalan*biasanya menimbulkan variasi ejaan dan pengecualian pada kata dasar. Kesalahan mungkin terjadi selama*berasal*proses dimana kata dasar tidak lengkap. Algoritma Idris memiliki aturan lain yang disebut Aturan 2 yang dapat diterapkan pada*awalan*penghapusan saja [15]. Aturannya adalah jika kata tersebut*berasal*belum lengkap, periksa huruf pertama kata tersebut dan jika kata tersebut diawali dengan huruf vokal, maka:

1. Tambahkan*T*setelah dihapus*laki-laki-*atau*pena-*
2. Tambahkan*k*setelah dihapus*meng-*atau*peng-*
3. Tambahkan*S*setelah menghapus*meny-*atau*sen-*
4. Tambahkan*F*atau*P*setelah dihapus*mem-*atau*pem-*

3. MODIFIKASI ALGORITMA IDRIS

Pada bagian ini, kami menjelaskan usulan modifikasi algoritma Idris, yang disebut IN-Idris. Dari hasil percobaan dan analisa stemmer Idris ada beberapa

contoh kata yang tidak tepat. Misalnya saja kata "*memasukkan*" (untuk masuk, dalam bahasa Inggris) menemukan akar kata "*pasuk*". Kata tersebut ada dalam kamus dan algoritme akan menganggapnya sebagai kata dasar, tetapi kata "*P bertanya*" tidak dimaksudkan, dan akar kata sebenarnya adalah "*M bertanya*" (masuk, dalam bahasa Inggris).

Hasil stemming kurang sesuai karena algoritma Idris terlebih dahulu menghilangkannya *awalan*menurut Aturan 2. Dalam contoh, setelah menghapus*awalan*"*mem*-"kata"*bertanya*" diperoleh, maka aturan keempat pada Aturan 2 setelah menghilangkan awalan "*mem*-" lalu tambahkan surat itu*P*, maka menjadi kata "*pasuk*". Eliminasi secara langsung mengacu pada efek Rule 2 pada hasil stemming, dimana algoritma tidak selalu menghasilkan kata sebenarnya.

Proses tersebut terjadi karena pada tahap ke-2, algoritma hanya memeriksa apakah kata-kata tersebut sesuai atau tidak*awalan*aturan dan pola Aturan 2 tanpa*awalan* penghapusan dan jika pada tahap ke 3 jika kata yang diperiksa sesuai dengan Aturan 2, algoritma segera menerapkan aturan tersebut pada kata tersebut. Setelah menganalisis hasil stemming, langkah selanjutnya adalah melakukan perbaikan dengan mengubah proses dari tahap 2 dan 3. Usulan perbaikan adalah sebagai berikut:

1. Pada langkah ke 2 algoritma akan memeriksa kata yang ada di dalam terlebih dahulu*awalan*aturan, jika sesuai maka*awalan*akan segera dihapus.
2. Pada langkah 3 jika memeriksa*awalan*aturan tidak sesuai, algoritma akan memeriksa *awalan* pola dalam Aturan 2 dan huruf pertama dari kata masukan. Jika sesuai dengan Rule 2 maka algoritma akan menerapkan rule tersebut pada kata tersebut.

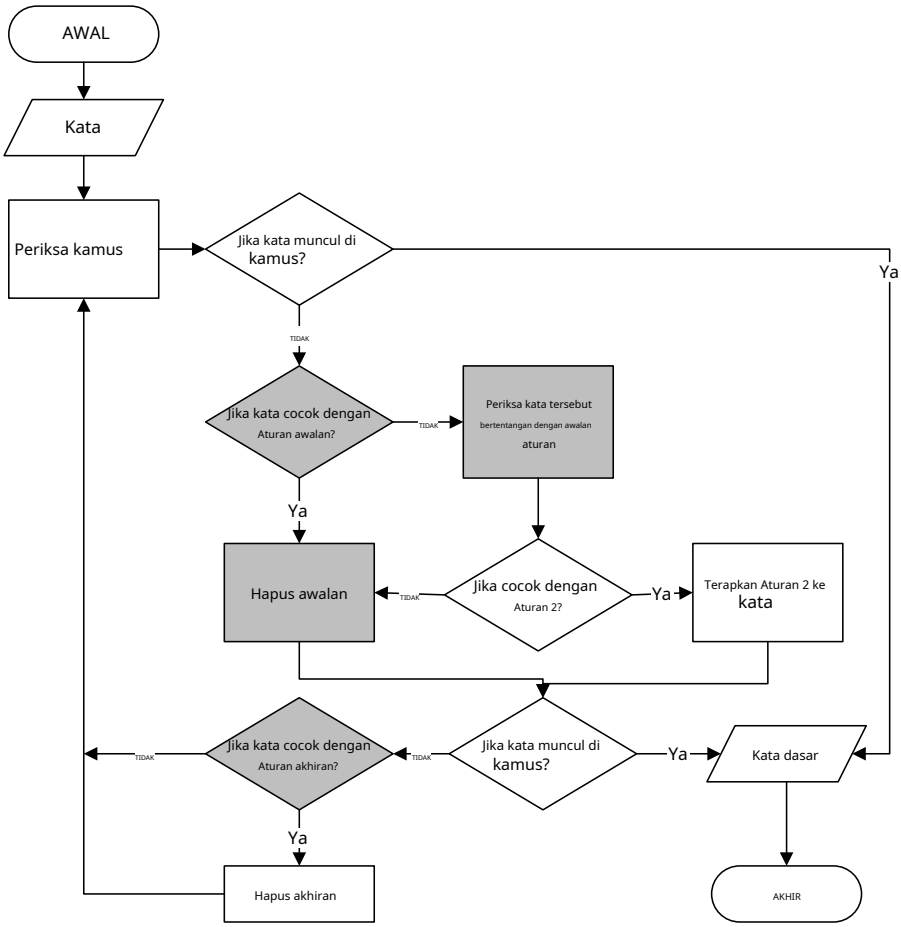
Modifikasi algoritma Idris dijelaskan pada Tabel 2 di bawah ini:

Tabel 2. Deskripsi algoritma IN-Idris

ALGORITMA 2 : Algoritma IN-Idris
1. Periksa kata tersebut di kamus. Jika kata tersebut ada dalam kamus, maka kata tersebut dianggap sebagai kata dasar dan kata keluar. Jika tidak, lanjutkan ke langkah berikutnya.
2. Periksa kata di*awalan*aturan. Jika kata tersebut cocok dengan aturan awalan, maka hapus*awalan*dan lanjutkan ke langkah 6. Jika tidak, lanjutkan ke langkah berikutnya.
3. Periksa*awalan*pola dan huruf pertama dari kata yang akan dibentuk. Jika*awalan* polanya cocok dengan Aturan 2, lalu terapkan Aturan 2 pada kata tersebut. Jika tidak, hapus*awalan*dan lanjutkan ke langkah 6.
4. Periksa*awalan*dari kata tersebut, sesuaikan polanya pada Aturan 2. Jika cocok dengan aturan keempat, maka periksa kata dasar di kamus dan lanjutkan ke langkah berikutnya. Jika tidak, hapus*awalan*dan lanjutkan ke langkah 6.
5. Jika kata tersebut tidak ditemukan dalam kamus, kembali ke langkah 4, jika tidak, hapus*awalan*dan lanjutkan ke langkah berikutnya.
6. Periksa kata tersebut di kamus. Apabila kata tersebut ditemukan, maka kata tersebut dianggap sebagai kata dasar dan kata keluar. Jika tidak, lanjutkan ke langkah berikutnya.
7. Periksa kata-kata di*akhiran*aturan. Jika cocok dengan*akhiran*aturan, lalu hapus*akhiran* dan lanjutkan ke langkah 1. Jika tidak, lanjutkan saja ke langkah 1.

Dari Gambar 1 kita dapat melihat langkah demi langkah dari algoritma yang diusulkan yang digambarkan melalui diagram alur. Proses dengan warna abu-abu menggunakan aturan modifikasi dari Idris stemmer.

https://doi.org/10.31436/iiumej.v23i1.1783



Gambar 1. Diagram alir algoritma Idris yang dimodifikasi (IN-Idris).

4. EKSPERIMEN DAN EVALUASI

4.1 Kumpulan Data

Untuk proses eksperimennya, penelitian ini menggunakan dataset yang diambil dari kutipan novel online Indonesia untuk mengukur kinerja algoritma yang diusulkan. Data novel bersumber dari penelitian Permatasi [6] yang berjumlah 6 novel, dan novel koleksi kami berjumlah 4 novel sehingga total ada 10 novel yang akan digunakan. Data kutipan novel direpresentasikan sebagai sebuah dokumen sehingga total dokumen tersebut adalah 10 dokumen untuk proses evaluasi. Data kutipan novel yang digunakan adalah isi saja tanpa menggunakan judul novel. Kumpulan data harus melalui *pemrosesan awal* tahap sehingga dapat diproses ke tahap selanjutnya. Sedangkan kamus akar kata yang digunakan berdasarkan Kamus Besar Bahasa Indonesia (Kamus Besar Bahasa Indonesia, dalam bahasa Inggris). Kamus Besar Bahasa Indonesia (KBBI) merupakan kamus bahasa Indonesia terlengkap yang populer digunakan sebagai referensi dalam pengolahan teks bahasa Indonesia.

4.2 Skenario Evaluasi

Evaluasi penelitian dilakukan dengan membandingkan *berasal* hasil modifikasi algoritma Idris (IN-Idris) dan lain-lain *berasal* algoritma (Idris asli, N&A, ECS). Dalam percobaan ini, kami menganalisis akurasi dan kecepatan setiap stemmer. Pengujian skenario yang dilakukan menguji 10 kutipan novel Indonesia. Output yang dihasilkan berupa kata dasar stemming yang dilanjutkan dengan menghitung jumlah hasil stemming benar dan salah dari masing-masing algoritma. Beberapa langkah digunakan,

https://doi.org/10.31436/iiumej.v23i1.1783

termasuk pelipatan kasus, tokenisasi, eliminasi *kata-kata penghenti*, Dan *berasal*. Case lipat adalah proses mengubah semua huruf menjadi huruf kecil dan menghilangkan tanda baca, angka, dan simbol tertentu [9]. Setelah itu, proses tokenisasi adalah memecah aliran teks menjadi frasa, kata, simbol, atau elemen lainnya [23]. Kemudian, kata-kata yang diambil dari hasil tokenisasi tersebut akan diproses untuk menghilangkan kata-kata tidak penting yang dimasukkan ke dalam *kata-kata penghenti* daftar. Selanjutnya setiap isi dokumen akan di stem. Itu *berasal* proses kemudian dilakukan untuk memproses *afiks* kata menjadi kata dasar.

Hasilnya akan dinilai untuk menghasilkan jumlah kata yang benar dan kata yang salah; sehingga nilai akurasi keduanya *stemmer* dapat dihitung. Evaluasi ini juga mengukur kecepatan waktu proses selama pergantian kata dasar dari kata dasar dokumen. Persamaan. (1) [8] digunakan untuk menghitung nilai akurasi *berasal* hasil dari setiap dokumen D_{Saya} Di mana *Saya*=1 sampai 10. Selanjutnya, *T* mewakili total kata-kata yang sebenarnya dari D_{Saya} ketika *w* adalah total kata-kata yang salah dari D_{Saya} dan kata total diwakili oleh *N*.

$$= (-) \ 100\% \quad \text{——} \tag{1}$$

4.3 Hasil Percobaan dan Pembahasan

Percobaan pertama membandingkan *berasal* hasil stemmer Idris dan IN-Idris. Hasil pengujian ditunjukkan pada Tabel 3 yang menjelaskan hasil percobaan untuk setiap dokumen yang diuji. Tabel tersebut juga menyajikan jumlah kata yang terdapat pada setiap dokumen, jumlah kata benar dan salah, serta nilai kecepatan dan ketepatan keduanya. *stemmer*. Baik untuk Idris maupun IN-Idris *stemmer*, akurasi maksimal dihasilkan dari dokumen 2 yang berubah dari 83,13% menjadi 87,95%. Sebaliknya, akurasi minimum dihasilkan pada dokumen 7 *berasal* hasil dengan sekitar 69,49% untuk IN-Idris dan 67,01% untuk Idris *batang*.

Tabel 3: Perbandingan akurasi dan kecepatan Idris dan IN-Idris Stemmer

Dokter	Nomor dari kata-kata	Idris				DI-Idris			
		BENAR Kata	Salah Kata	Kecepatan (S)	Ketepatan (%)	BENAR Kata	Kecepatan Kata	Salah (S)	Ketepatan (%)
1	239	97	27	3.40	78.23	106	17	3.32	86.18
2	261	120	28	3.62	81.08	127	21	3.02	85.81
3	323	138	28	3.79	**83.13**	146	20	3.80	**87,95**
4	511	188	46	**5.28**	80.34	192	42	**5.96**	82.05
5	608	233	79	7.96	74.68	257	55	8.81	82.37
6	2224	912	230	**42.16**	79,86	967	173	**47.17**	84.82
7	1361	522	257	25.56	**67.01**	539	237	18.52	**69.46**
8	1210	415	137	14.71	75.18	443	109	13.59	80.25
9	794	361	91	12.95	79,87	384	65	12.90	85.52
10	997	366	114	18.61	76.25	400	78	18.40	83.68
Rata-rata				**13.80**	**77.56**			**13.55**	**82.81**

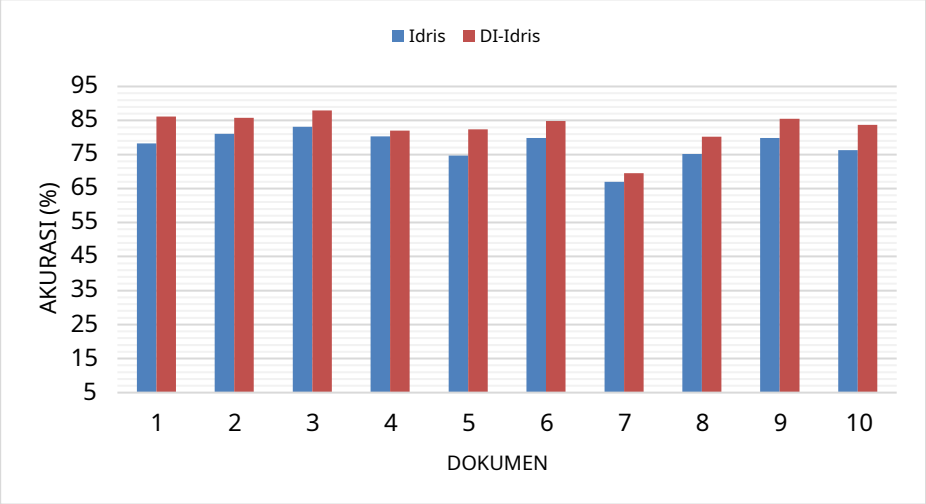
Sebaliknya, dalam hal kecepatan, *berasal* berdasarkan IN-Idris dibutuhkan sekitar 13,55 detik. Namun, ini lebih cepat dibandingkan memproses kata dasar dengan Idris *batang* yang memiliki waktu rata-rata sekitar 13,80 detik. Secara umum, berdasarkan percobaan peneliti (ditunjukkan pada Tabel 3), dapat disimpulkan bahwa dokumen yang memiliki kata di bawah 600 hanya membutuhkan waktu kurang dari 9 detik untuk *berasal*. Sedangkan jika jumlah kata-katanya adalah

https://doi.org/10.31436/iiumej.v23i1.1783

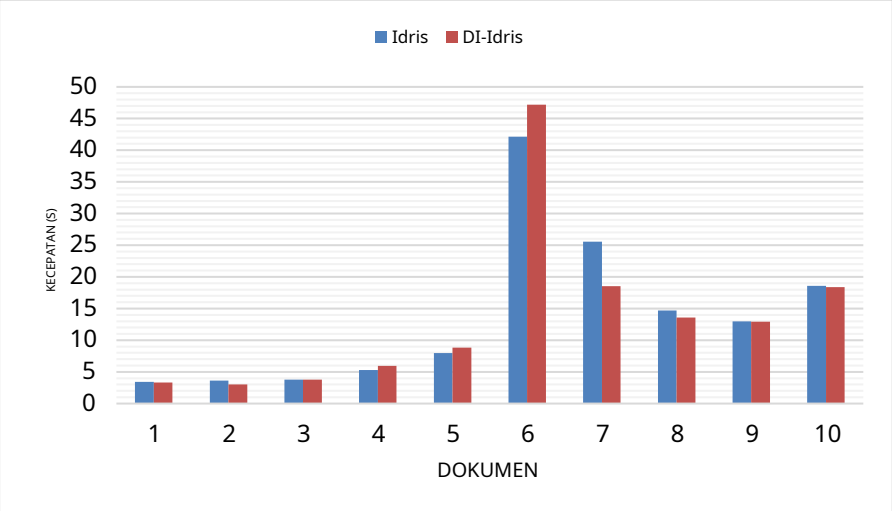
di bawah 1300, prosesnya akan memakan waktu kurang dari 25 detik. Selain itu, untuk dokumen yang memiliki lebih dari 2000 kata, diperlukan waktu lebih dari 47 detik.

Gambar 2 dan 3 menunjukkan perbandingan akurasi dan kecepatan dari keduanya *stemmer* direpresentasikan dalam sebuah grafik. Berdasarkan Gambar 2 secara umum menunjukkan bahwa nilai akurasi yang dihasilkan IN-Idris lebih tinggi dibandingkan dengan Idris *batang* hasil. Nilai rata-rata akurasi dari IN-Idris *batang* adalah sekitar 82,81% sementara *berasal* menggunakan algoritma Idris menunjukkan akurasi hanya 77,56%. Kinerja IN-Idris mendominasi seluruh dokumen yang diuji dan mencapai akurasi lebih dari delapan dari sepuluh.

Secara umum, IN-Idris menyajikan kecepatan yang lebih cepat dibandingkan Idris dalam pengolahan kata dasar (ditunjukkan pada Gambar 3). Namun Idris menunjukkan kecepatan yang baik dalam memproses dokumen 4, 5, dan 6. Dari segi akurasi, IN-Idris lebih baik dibandingkan Idris dalam memproses dokumen tersebut. Kecepatan rata-rata algoritma yang diusulkan mencapai kurang lebih 13,55 detik sedangkan Idris membutuhkan rata-rata 13,80 detik untuk *berasal*. Akibatnya terjadi kesenjangan kecepatan antara IN-Idris dan Idris *stemmer* adalah sekitar 0,25 detik.



Gambar 2. Visualisasi grafik perbandingan akurasi antar Idris dan IN-Idris *berasal*.



Gambar 3. Visualisasi grafik kecepatan antara Idris dan IN-Idris *berasal.*

https://doi.org/10.31436/iiumej.v23i1.1783

Tabel 4 menunjukkan contoh teks dari *berasal* hasil dari Idris dan IN-Idris *stemmer*. Tabel tersebut berisi: kata dasar (kata asli yang diambil dari kumpulan novel), kata dasar kebenaran dasar (kata yang dijadikan acuan penilaian), dan kata dasar *berasal* hasil dari kedua stemmer. Itu *berasal* Kolom hasil menampilkan kata dasar mana yang berhasil diubah menjadi kata dasar dan juga menunjukkan kata dasar mana yang tidak berhasil. Ada beberapa kasus dimana IN-Idris lebih berkuasa dibandingkan Idris, terbukti dari kasus no. 1, 2, 4, 5, 6, 7, 9, dan 10. Namun untuk kasus no. 3 kedua algoritma tidak dapat bekerja dengan baik. Beberapa kasus menunjukkan bahwa Idris *batang* pengurangan karakter yang terpengaruh, seperti kata dasar "pendek" diubah menjadi "dek", dan kata dasar "tetes" diubah menjadi "tes".

Tabel 4: Contoh teks dari *Berasal* hasil

Kasus TIDAK.	Kata Batang	Akar Kata Kebenaran dasar	Dalam bahasa Inggris	Hasil Steming IDRIS	DI-IDRIS
1	*menyatakan*	*musnah*	hancur	*menyatakan*	*musnah*
2	*percayai*	*percaya*	meyakini	*percayai*	*percaya*
3	*kelahiranmu*	*lahir*	dilahirkan	*kelahiranmu*	*kelahiranmu*
4	*menakutkan*	*seram*	menakutkan	*menakutkan*	*seram*
5	*menyakitkan*	*sakit*	sakit	*menyakitkan*	*sakit*
6	*mengerikan*	*ngeri*	ngeri	*gerik*	*ngeri*
7	*pendekku*	*pendek*	pendek	*dek*	*pendek*
8	*terulur*	*ulur*	berbaring	*ulut*	*ulut*
9	*tetesan*	*tetes*	tetes	*tes*	*tetes*
10	*menuruni*	*turun*	turun	*urun*	*turun*

Pada percobaan selanjutnya, kami membandingkan hasil IN-Idris *batang* dengan dua algoritma lain (N&A dan ECS) selain hasil Idris sebelumnya. Tabel 3 menyajikan akurasi dan kecepatan keempatnya *batang* hasil saat diuji menggunakan 10 dokumen terpilih. Dari tabel, keakuratan ECS *batang* pada akurasi atas dan bawah masing-masing sekitar 90,37% dan 80,36%. Di sisi lain, hasil N&A *batang* memiliki akurasi maksimum sebesar 89,86%, dan akurasi minimum hanya 73,78%. Sedangkan akurasi Idris dan IN-Idris telah dijelaskan dan disajikan pada tabel sebelumnya. Secara keseluruhan, hasil akurasi menunjukkan kinerja yang signifikan pada delapan dari sepuluh.

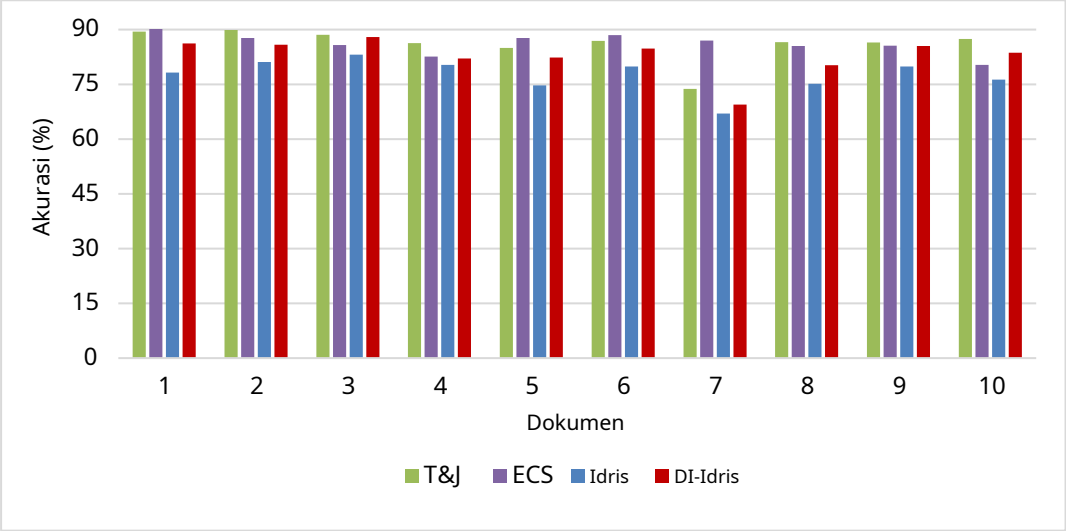
Dari segi kecepatan, Tabel 5 menggambarkan waktu tercepat yang dibutuhkan oleh ECS *batang* adalah sekitar 2,41 detik sedangkan waktu terlama sekitar 30,50 detik. Hasil ini tidak jauh berbeda dengan hasil N&A yang mempunyai kecepatan maksimum dan minimum masing-masing sekitar 36,68 detik dan 2,51 detik. Sedangkan Idris memiliki kecepatan terendah sekitar 42,16 detik dan kecepatan tercepat sekitar 3,40 detik. Sedangkan IN-Idris memiliki kecepatan tercepat dan terpanjang masing-masing sekitar 3,02 detik dan 47,17 detik.

Pada Gambar 4 disajikan grafik perbandingan akurasi keempatnya *berasal* algoritma. Secara keseluruhan, hasil akurasi menunjukkan kinerja yang signifikan pada delapan dari sepuluh. Sedangkan Idris menghasilkan akurasi yang lebih rendah, yaitu ketidakakuratan hanya mencapai 77,56%, dan ECS *batang* memiliki akurasi tertinggi sebesar 85,92%. Sementara itu, diusulkan *batang,* IN-Idris, menunjukkan akurasi sebesar 82,81%, lebih rendah dibandingkan ECS dan N&A *stemmer*, namun kesenjangan akurasinya kecil, hanya sekitar 3%.

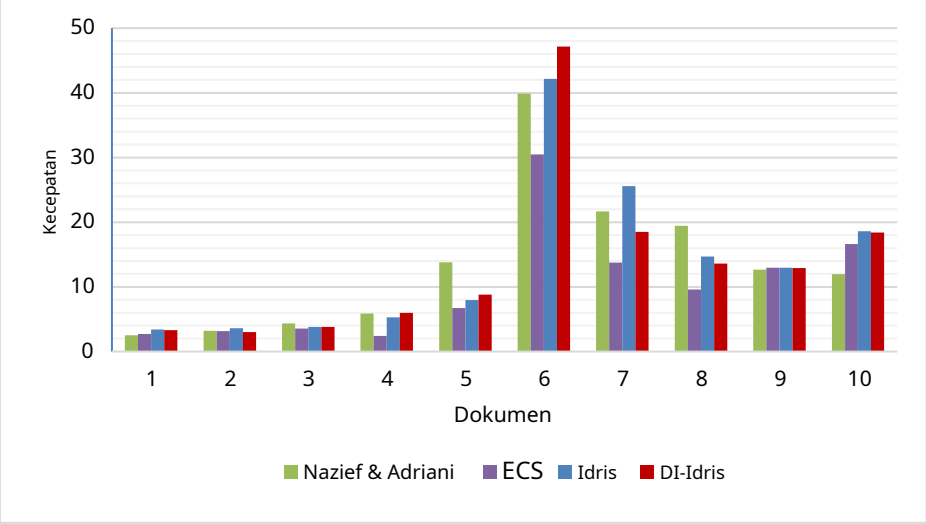
https://doi.org/10.31436/iiumej.v23i1.1783

Tabel 5: Hasil perbandingan akurasi dan kecepatan antara Idris dan IN-Idris*batang*

Dokter	T&J		ECS		Idris		DI-Idris	
	Kecepatan (S)	Ketepatan (%)	Kecepatan (S)	Ketepatan (%)	Kecepatan (S)	Ketepatan (%)	Akurasi Kecepatan (S)	(%)
1	2.51	89.43	2.73	90,37	3.40	78.23	3.32	86.18
2	3.19	89,86	3.14	87.73	3.62	81.08	**3.02**	85.81
3	4.37	88,55	3.55	85,75	3.79	83.13	3.80	87,95
4	5.87	86.32	2.41	82.58	5.28	80.34	5.96	82.05
5	7.35	84,94	6.71	87.66	7.96	74.68	8.81	82.37
6	36.68	86,93	30.50	88.46	42.16	79,86	47.17	84.82
7	21.66	73,78	13.73	87.00	25.56	67.01	18.52	69.46
8	19.45	86,59	9.59	85.49	14.71	75.18	13.59	80.25
9	12.65	86.49	12.98	85.61	12.95	79,87	12.90	85.52
10	11.97	87.45	16.62	80.36	18.61	76.25	18.40	83.68
Rata-rata	**12.57**	**85.61**	**10.20**	**85,92**	**13.80**	**77.56**	**13.55**	**82.81**



Gambar 4: Visualisasi grafik hasil perbandingan yang akurat untuk kedua stemmer.



Gambar 5. Visualisasi grafik hasil perbandingan akurat kedua stemmer.

Gambar 5 menggambarkan perbandingan*berasal*kecepatan keempat algoritma. *Berasal* diproses oleh ECS adalah yang tercepat, hanya 10,20 detik, dan terlama *berasal*dibutuhkan kurang lebih 13,80 detik bila diproses melalui Idris *batang*. Sedangkan metode yang diusulkan IN-Idris memiliki durasi waktu sekitar 13,55 detik untuk mengubah kata menjadi akarnya. Hasil ini menjadi bukti bahwa IN-Idris membutuhkan waktu ekstra dibandingkan N&A dan ECS dalam*berasal*. Namun, ada satu kejadian ketika IN-Idris tamat*berasal*lebih cepat dari yang lain, hal ini ditunjukkan untuk dokumen 2, dimana hanya membutuhkan waktu 3,02 detik.

Berdasarkan percobaan, nilai akurasi dipengaruhi oleh jumlah kata sebenarnya yang dihasilkan untuk setiap dokumen*berasal*hasil. Semakin tinggi jumlah kata benar maka semakin tinggi akurasi yang dihasilkan. Namun, semakin sedikit jumlah kata yang benar, semakin kecil pula akurasi yang dihasilkan. Selanjutnya, hasil dari*berasal*keakuratan tiap dokumen berbeda-beda tiap dokumennya*batang*. Hal ini dapat disebabkan oleh perbedaan aturan yang diterapkan pada setiap algoritma, hal ini dapat disebut a*batang*ciri.

Sebaliknya dari segi kecepatan, ECS*batang*lebih cepat dari algoritma Idris. Hasil tersebut merupakan bantahan terhadap penelitian sebelumnya [6] yang dilakukan untuk menguji keakuratan Idris dan ECS.*stemmer*diimplementasikan dalam dokumen teks berbahasa Indonesia. Penelitian tersebut mengklaim bahwa Idris memiliki performa yang sangat baik dalam hal kecepatan. Kami menyimpulkan bahwa kecepatan pemrosesan masuk*berasal*berkaitan dengan jumlah kata yang terdapat pada setiap dokumen, dimana waktu yang dibutuhkan untuk melakukan hal tersebut*berasal* Prosesnya relatif berdasarkan alat atau bahasa pemrograman yang digunakan [24].

Ditemukan bahwa algoritma Idris memiliki nilai akurasi yang rendah yang disebabkan oleh*afiks* kesalahan pemotongan di*berasal*proses yang mempengaruhi hasil. Kesalahan itu terkait dengan penerapan*awalan*penghapusan aturan pertama pada Idris*berasal*. Hal ini menyebabkan akar kata memiliki huruf yang sama dengan a*awalan*untuk dibuang, dan menyebabkan ketidakcocokan dan kesalahan pada hasil akhir. Kesalahan ini dapat diperbaiki oleh IN-Idris yang memperoleh hasil baik *berasal* hasil dengan peningkatan akurasi hingga 5,25% dari Idris*batang*. Namun pada modifikasi algoritma Idris, IN-Idris masih mempunyai kekurangan yang menyebabkan kata-kata gagal masuk *berasal*. Ada kekurangan yang tidak bisa diatasi oleh IN-Idris, seperti*afiks*"*se- ku*" pada kata "*sebelahku* " (di sebelah saya, dalam bahasa Inggris) sedangkan dengan algoritma Idris, kata "*sebelahku*" diubah menjadi "*bagian*".

Tantangan untuk mengembangkan suatu kebaikan*berasal*algoritma untuk pekerjaan masa depan adalah bagaimana a *batang*dapat mengatasi beberapa permasalahan diantaranya kesalahan penghapusan terutama yang terjadi karena kesalahan pengetikan dan kesalahan nama tempat, nama orang, dan juga bahasa asing. Yang paling kuat*batang*adalah algoritma ECS, dan beberapa aturannya dapat diadopsi untuk meningkatkan kinerja IN-Idris. Namun, ECS*batang*menerapkan *akhiran*aturan penghapusan terlebih dahulu dan memeriksa kamus setiap kali menghapus*afiks*. Hal ini dapat mengarah pada perkataan*berlebihan*karena*afiks*Proses penghapusan dilakukan semaksimal mungkin sesuai aturan yang berlaku, sehingga mempengaruhi keakuratan hasil. Selain itu, perlu adanya pengembangan lebih lanjut yang dapat memodifikasi kombinasi tersebut*awalan*Dan*akhiran* aturan untuk menghasilkan yang lebih baik*berasal*pertunjukan.

5. KESIMPULAN DAN PEKERJAAN MASA DEPAN

IN-Idris adalah orang baru*batang*untuk teks bahasa indonesia yang terinspirasi dan disempurnakan dari bahasa melayu idris*batang*. Berdasarkan percobaan yang dilakukan peneliti menemukan bahwa algoritma Idris memiliki nilai akurasi yang rendah yang disebabkan oleh kesalahan pemotongan imbuhan pada*berasal* proses yang mempengaruhi hasil. Kesalahan ini bisa diperbaiki oleh IN-Idris sehingga bisa

memperoleh kebaikan *berasal* hasil dengan peningkatan akurasi hingga 5,25% dari Idris *batang*. Performa IN-Idris mendominasi seluruh dokumen yang diuji, dan secara keseluruhan hasil akurasi menunjukkan performa signifikan di atas delapan dari sepuluh. Dari segi kecepatan, IN-Idris menghadirkan kecepatan yang lebih cepat dibandingkan Idris dalam pengolahan kata dasar. Terakhir, kami menyimpulkan bahwa nilai akurasi dipengaruhi oleh jumlah kata sebenarnya yang dihasilkan untuk masing-masing kata *berasal* dokumen hasil. Semakin tinggi kata benar maka semakin tinggi akurasi yang dihasilkan. Namun, semakin kecil kata sebenarnya, semakin kecil pula keakuratan yang dihasilkan.

REFERENSI

[1] Vijayarani DS, Ilamathi, MJ., Nithya M. (2015) Teknik praproses untuk penambangan teks - Gambaran umum. J. Ilmu Komputer & Jaringan Komunikasi, 5(1): 7-16.
[2] Buntoro G, Arifin R, Syaifuddiin G, Selamat A, Krejcar O, Hamido F. (2021) Implementasi algoritma pembelajaran mesin untuk analisis sentimen pemilu Presiden Indonesia 2019. Jurnal Teknik IIUM, 22(1): 78-92.
[3] Nassirtoussia AK, Aghabozorgia S, Wah TY, David CLN. (2014) Penambangan teks untuk prediksi pasar: Tinjauan sistematis. Sistem Pakar dengan Aplikasi, 7653-7670.
[4] Rizki AS, Tjahyanto A, Trialih R. (2019) Perbandingan algoritma stemming pada pengolahan teks bahasa Indonesia. Telkomnika, 17(1): 95-102.
[5] Utomo FS, Suryana N, Sanusi Azmi, M. (2020) Analisis Dampak Stemming Terjemahan Al-Quran Indonesia dan Klasifikasi Tafsirnya untuk Instansi Ontologi. Jurnal Teknik IIUM, 21(1): 33-50.
[6] Permatasari N. (2016) Analisis perbandingan algoritma Idris dan algoritma Enhanced Confix Stripping (ECS) stemmer pada dokumen teks bahasa Indonesia. Universitas Komputer Indonesia. https://repositori.unikom.ac.id/130/
[7] Titin W, Kerami J, Arief S. (2017) Penentuan Term pada clustering dokumen teks menggunakan algoritma Enhanced Confix Stripping Steming. Jurnal Internasional Aplikasi Komputer, 157(9): 8-13.
[8] Prasidhatama A, Suryaningrum KM. (2018) Perbandingan algoritma Nazief & Adriani dengan algoritma Idris Untuk pencarian kata dasar. Jurnal Teknologi & Manajemen Informatika, 4(1): 1-4.
[9] Prihatini PM, Putra ID, Giriantari IAD, & Sudarma M. (2017) Algoritma Stemming untuk Pengolahan Teks Berita Digital Indonesia. Jurnal Internasional Teknik dan Teknologi Berkembang, 2(2): 1-7.
[10] Mena VV, Saputri K. (2018) Analisis kontras antara prefiks dan sufiks bahasa Inggris dan Indonesia dalam teks deskriptif buku teks siswa. Jurnal Komunitas Bahasa Inggris, 2(1): 175-182.
[11] Jelita A. (2007) Teknik Efektif Pengambilan Teks Bahasa Indonesia. Melbourne: Universitas RMIT.
[12] Adriani M, Asian J, Nazief B, Tahaghoghi SMM, & Williiams HE. (2007) Stemming Bahasa Indonesia: Pendekatan confix-stripping. Transaksi ACM pada Pemrosesan Informasi Bahasa Asia (TALIP), 6(4): 1-33.
[13] Widayanto H & Huda AF. (2017) Perbandingan algoritma stemmer Nazief Adriani dan CS untuk data nyata stemm. Dalam e-Prosiding Teknik, 4(3): 5215. Bandung, Indonesia.
[14] Arifin AZ, Mahendra IPAK, & Ciptaningtyas HT. (2009) Peningkatan algoritma Confix stripping stemmer dan ants untuk mengklasifikasikan dokumen berita berbahasa Indonesia. dalam Konferensi Internasional tentang Teknologi dan Sistem Informasi & Komunikasi. Di dalam Konferensi Internasional tentang Teknologi dan Sistem Informasi & Komunikasi, 5:149-158. Surabaya, Indonesia.
[15] Idris N, Mustafa SS. (2001) Berasal dari Penggabungan Istilah Dalam Teks Melayu.
[16] Porter, MF (1980). Algoritma untuk pengupasan sufiks. Program1, 14(3): 130-137.
[17] Paramita ES. (2012) Analisis dan implementasi stemming menggunakan algoritma Idris pada dokumen teks berbahasa indonesia. Universitas Telkom, Indonesia.

[18] Mardiana T, Adji TB, Hidayah I. (2016) Pengaruh Stemming Terhadap Deteksi Kemiripan Abstrak yang Ditulis di Indonesia. Telkomnika, 14(1): 219-227.
[19] Rifai W, Winarko E. (2019) Modifikasi algoritma Stemming menggunakan pendekatan non deterministik pada teks bahasa Indonesia. Jurnal Sistem Komputasi dan Sibernetika Indonesia, 13(4): 379-388.
[20] Vega VB. (2001) Temu kembali informasi untuk bahasa Indonesia. Tesis master, Universitas Nasional Singapura.
[21] Arifin AZ & Setiono AN. (2002) Klasifikasi dokumen berita kejadian berbahasa Indonesia dengan algoritma single pass clustering. Dalam Prosiding Seminar Teknologi Cerdas dan Penerapannya (SITIA). Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.
[22] Asia J, Williams HE, & Tahaghoghi SMM. (2005) Membendung Bahasa Indonesia. Dalam Prosiding konferensi Australasia ke-28 tentang Ilmu Komputer, 38: 307-314. Australia.
[23] Winarti T, Kerami J, Arief S. (2014) Proses Tokenisasi dan Filtering di rapidminer. Jurnal Internasional Sistem Informasi Terapan, 7(2): 16-18.
[24] Zaman B. (2014) Modifikasi algoritma Porter untuk stemming pada kata bahasa Indonesia. Dalam Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA 2014), 543-550. Surabaya, Indonesia.